Digital Investigation 11 (2014) S57–S65

Contents lists available at ScienceDirect

Digital Investigation

journal homepage: www.elsevier.com/locate/diin

Preliminary forensic analysis of the Xbox One

Jason Moore [a], Ibrahim Baggili [a,*], Andrew Marrington [b], Armindo Rodrigues [a]

[a] Cyber Forensics Research and Education Group (UNHcFREG), University of New Haven, Tagliatela College of Engineering, ECECS Department, 300 Boston Post Rd, West Haven, CT 06416, United States
[b] Advanced Cyber Forensics Research Laboratory, Zayed University, PO Box 19282, Dubai, United Arab Emirates

Keywords:
Xbox one
Video game console forensics
Network forensics
Games
Digital forensics
NTFS
Hard drive

A B S T R A C T

Video game consoles can no longer be viewed as just gaming consoles but rather as full multimedia machines, capable of desktop computer-like performance. The past has shown that game consoles have been used in criminal activities such as extortion, identity theft, and child pornography, but with their ever-increasing capabilities, the likelihood of the expansion of criminal activities conducted on or over the consoles increases. This research aimed to take the initial step of understanding the Xbox One, the most powerful Microsoft console to date. We report the outcome of conducting a forensic examination of the Xbox One, and we provide our Xbox One data set of hard drive images and unique files so that the forensic community may expand upon our work. The Xbox One was found to have increased security measures over its predecessor (Xbox 360). The encryption of the data and the new file types introduced made it difficult to discern potential digital evidence. While these added security features caused great difficulty in forensically acquiring digital forensic artifacts, some important and interesting digital evidence was gathered using open-source tools. We were able to find digital evidence such as times that the user initially set up the console, and times when the system was restored or shutdown. We were also able to determine what games and applications had been downloaded along with when the games were played. Finally, through our network forensic experiments, we were able to determine that various applications had different levels of security and that game traffic was encrypted.

Introduction

According to Forbes, as of January 2014, Microsoft had sold approximately 3.4 million Xbox One units since its release on November 22, 2013. In recent years, Microsoft's Xbox systems have maintained top gaming console sales amongst its competitors.

Gaming systems are now comparable to desktop computers such that they are just as powerful, have networking capabilities, they contain high-powered graphics processors and a large amount of storage.

As the capabilities of these gaming consoles increase, so does the potential for them to be used in illicit activities. Criminal investigators have historically sought to gather evidence from PCs, mobile phones, PDAs, and other mobile devices; however, they may overlook these gaming consoles even though they can potentially contain valuable forensic artifacts that can be used for evidentiary purposes. It is imperative to provide analysis of the Xbox One to provide investigators an understanding of the proprietary system with hopes of retrieving the evidence it may hold.

This research serves as an initial examination of the Xbox One and its file system, allowing investigators some information on how to investigate an Xbox One. This includes understanding the file structure of the system, what

* Corresponding author. Tel.: +1 203 932 7198.
E-mail address: IBaggili@newhaven.edu (I. Baggili).

http://dx.doi.org/10.1016/j.diin.2014.05.014
1742-2876/

forensically valuable information is available on the console, and where that information may be located.

Contribution

With the more advanced functionality, power, and complexity that the Xbox One holds over the previous generations of consoles, it is important to understand where to find digital evidence on the hard drive and how to retrieve it. To the best of our knowledge, this research is the first research-centric digital forensic investigation of an Xbox One, which aims to provide a base for which future works on the device.

In addition to the research provided, we release a data set containing the hard drive images along with exported files that have a unique, unknown file types. This will provide the forensic community with a data set that allows further digital forensic research on the Xbox One. This data set will include the hard drive images from Phase I and Phase II as well as various files found on the device. It can be requested at http://www.unhcfreg.com <Data & Tools> and will be made available upon the request and identity verification of the researchers.

Literature review

Gaming consoles and criminal activity

As the popularity of gaming consoles increases, so do the instances in which a console is used to commit a crime. Criminals often hide illicit data on gaming consoles such as the Xbox, Xbox 360, Sony PlayStation 3 and 4, and the Nintendo Wii in hopes that the console will not be perceived as a likely evidence target, especially when personal computers have been seized as evidence (Collins, 2009). Consoles have been known to harbor fraudulent documents, illegal software, indecent images, etc., and have been used to steal identities (Vaughan, 2004; Conrad et al., 2009; Podhradsky, Sley, D'Ovidio, & Casey, 2011). Other notable cases don't involve just storage of illicit material, but the crimes are actually committed over the gaming network. A known case in this regard involves using Xbox Live to exploit children (Bolt, 2011).

Forensic analysis of a Sony PlayStation 3 gaming console

The PlayStation 3's operating system was found to be encrypted despite providing the option to install and run a secondary operating system. Furthermore, of great importance in this published study was the finding that digital forensic acquisition tools were able to recognize partitions and identify the file system, however the tools were not able to actually read the files and displayed said files as unpartitioned space (Conrad et al., 2009).

Xbox 360

While the video game console forensic field is still in its infancy, some work has been conducted on the newer consoles, albeit, the information is limited and still needs to be expounded upon. These works have aided with highlighting the importance of forensically analyzing gaming consoles for evidentiary purposes, while also providing awareness to the fact that a video game console is now being used as a media hub and for using the Internet in various ways.

It has been shown that an unmodified Xbox 360 can provide interesting digital forensic artifacts that are useful in a criminal investigation including alibies, online presence, and activities whether innocent or illegal. Log files saved to the hard drive identified the timestamps of gaming sessions including the time, date, and length of a gaming session, the gamertags (online user names) of the online players during interactive sessions, and network activity (Xynos et al., 2010).

Modding

An 'out-of-the-box' Xbox or Xbox 360 only allows code to be run that is authorized by Microsoft. Therefore, conducting activity that is outside the scope of what Microsoft had envisioned for their system, i.e. saving personal data to the hard drive, was not possible. However, the system was quickly hacked by modders allowing the system to be used as if it was a personal computer. These modifications typically allow users to run alternate operating systems (typically Linux based), execute illegal software, save personal data, conceal partitions, etc. Many tools, along with websites and comprehensive guides, were developed allowing even the average user access to these modifications (Modfreakz, 2009).

Methodology and tools

The methodology and tools used in this research followed the guidelines for forensically examining artifacts as deemed by NIST (Kent et al., 2006).

This research was separated into three separate phases:

- Phase 1 – The Xbox One was restored to factory settings. The hard disk drive was then removed from the system and forensically imaged while using a hardware write-blocker. Various methods of analysis of the hard drive were followed.
- Phase II – The hard drive was reinstalled into the system and the staged events below were conducted. Once all the events were completed, imaging and analysis were performed as in Phase I:
  - Installed Battlefield 4, played in both multiplayer and single player modes.
  - Installed Dead Rising 3, played in both multiplayer and single player modes.
  - Installed and used various apps, which consisted of: Skype, Twitch, YouTube, Xbox Video, Xbox Music, and FXNow.
  - A cable box was hooked up through the Xbox One to allow television to be viewed through the console.
  - The user signed in using the facial recognition feature.
  - The user signed in without using the facial recognition feature.
  - Viewed friend's gameplay videos.
- Phase III – Since the Xbox One would undoubtedly be used in an online environment; some analysis of the interaction between the Xbox One and the Internet was

required. The following controlled events were used to examine this communication:

- Used the YouTube, Skype, Internet Explorer, Twitch, and Game DVR applications.
- Played Battlefield 4 in single player and on the Xbox Live network with other users.
- Played Dead Rising 3 in single player and on the Xbox Live network with other users.
- Signed in and out of the user's profile.

Image acquisition

Forensic guidelines state that the copies of the source drive must be taken in a manner such that any potential evidence present on the system will experience minimum alteration. The Xbox One came with a SATA hard disk drive which was removed from the system and connected to a Forensics Recovery of Evidence Device (FRED) to be imaged using AccessData's FTK Imager (*AccessData, 2013*). FRED contains a built-in write-blocker, which was validated before the imaging process was started. Hashes of the hard drive were taken before and after imaging to ensure that no data on the drive was modified during the imaging process so that a forensically sound image of the hard disk was created.

Autopsy

After the image was acquired, Autopsy was used to preliminarily examine at the data on the hard disk drive and perform a keyword search (Carrier, 2013). The keywords were derived from the procedures of the experimental scenario such as the names of the games played, the names of the researchers, the string of words used in an email, and the dates and times the system was used.

Data carving

Data carving involves reconstructing files based on their content, rather than the metadata that points to the content (Garfinkel, 2007). It is an important technique in digital forensics allowing investigators to recover data that may otherwise be lost. There are many forensic tools available to accomplish this task, two open-source tools with a strong reputation were chosen for this research – bulk_extractor and Scalpel.

bulk_extractor is a forensic tool that scans a disc image and extracts information without parsing the file system or file system structure (Garfinkel, 2012). It is capable of processing different areas of the disk in parallel, does not miss data in unallocated regions of file systems and can process any type of digital media, making bulk_extractor fast, thorough, and flexible.

Scalpel is a high performance file carver that is based on three primary requirements: i) Frugality – allowing it to run on machines with low resources, ii) High performance – making it able to perform carving as fast as possible, and iii) Support for distributed implementation – giving it the ability to be adaptable to a distributed cluster based digital forensics platform (Richard III & Roussev, 2005).

Network analysis

To examine the packets being sent to and from the Xbox One, a shared Internet connection was established between a host computer, which had Internet access, and the Xbox One device. This was accomplished by connecting the Xbox One via an Ethernet cable to the host computer, which then provided the Xbox One with an Internet connection.

Two tools were chosen to carry out network acquisition and analysis; Wireshark and NetworkMiner. Wireshark is a network packet analyzer that captures packets and displays them with as much detail as possible (Combs, 2013). NetworkMiner is used to detect operating systems, sessions, hostnames, open ports etc., and has the ability to reconstruct data in PCAP files to reassemble transmissions (NETRESEC, 2013).

Analysis

Partition layout

The Xbox One's hard disk contained the five NTFS partitions shown in Table 1.

Factory restoration of the console took place at 8:50 PM UTC on 1/20/2014 (the Xbox One records timestamps in UTC). This timestamp was carried by the NTFS metadata structures found on every NTFS partition. This research took place in the Eastern Standard Time (EST) zone; therefore the timestamps were all five hours ahead of the local time. For consistency with the figures throughout the document, all of the times in this paper will be in UTC.

Preliminary results

Data carving with bulk_extractor and Scalpel proved to be ineffective, likely due to the encryption and/or compression, which we assume to be in use on the Xbox One (see below). Scalpel was able to carve a multitude of files but they could not be opened or viewed – indicating that the files were carved incorrectly. Likewise, bulk_extractor extracted a large quantity of text from the volume, but most was not human readable. This left us to parsing the MFT to draw conclusions and locate potential digital evidence, and to analyze the files we could find from a logical view of the file system. The files we could locate were of unknown binary file types, and were not human readable. Most files had the extension "xvd", but this is no guarantee that they share the same file type. In order to speculate as to the nature of these files, we calculated an entropy score for each file, applying the Forensic Relative

Table 1
Xbox One partitions.

Partition	Size (MB)
Temp Content	41,984
User Content	373,760
System Support	40,960
System Update	12,288
System Update 2	7168
Unpartitioned Space	GPT

Strength Scoring approach to explore the nature of the unknown file types (Shannon, 2004). The entropy score for each file was expressed as the minimum number of bits needed to encode each byte of information in the most optimal compression regime. Human readable text formats tend to score between 3 and 5 bits per byte, while encrypted or compressed files tend to score between 7 and 8 bits per byte (Shannon, 2004).

Master file table

The MFT is the most important feature of NTFS. The MFT includes all of the information about the files on the system; there is at least one entry in the MFT for every file on an NTFS file system volume, including the MFT itself. The data within a MFT contains file metadata, information that could be very helpful for investigators. Metadata is the data in the file system that describes the layout and attributes of the files and directories, i.e., timestamps, file size, etc. (Buchholz and Spafford, 2004). This can assist investigators in determining timelines, patterns of use, suspicious files, etc.

The MFT of each partition was parsed using a program known as mft2csv, an open-source tool. An $MFT file is taken as an input, information is extracted from the $MFT records and is logged it to a comma-separated values (CSV) file (Schicht, 2014).

The following sections will discuss the information that was discovered on each of the system's partitions.

Temp content partition

Figs. 1 and 2 highlight the content of the root directory in the temp content partition both before and after use.

There were several areas of interest in this partition. By calculating MD5 hash digests for files from both the "before" and "after" images, we noticed that these files experienced modifications during the usage scenario: *appswapfile.xvd, AppTempStorage, $sosrst.xvd AppUserStorage, ConnectedStorage-retail,* and *GDVRIndex.xvd.*

Of the abovementioned six files, all were created when the system was restored to factory settings. Interestingly however, in Phase II, *$sosrst.xvd* and *appswapfile.xvd* had file creation timestamps of 14:41:33 and 14:41:37

Name	Size	Type	Date Modified
$Extend	1	Directory	1/20/2014 8:50:36 PM
$AttrDef	3	Regular File	1/20/2014 8:50:36 PM
$BadClus	0	Regular File	1/20/2014 8:50:36 PM
$Bitmap	1,312	Regular File	1/20/2014 8:50:36 PM
$Boot	8	Regular File	1/20/2014 8:50:36 PM
$I30	4	NTFS Index All...	1/20/2014 8:50:36 PM
$LogFile	65,536	Regular File	1/20/2014 8:50:36 PM
$MFT	256	Regular File	1/20/2014 8:50:36 PM
$MFTMirr	4	Regular File	1/20/2014 8:50:36 PM
$Secure	1	Regular File	1/20/2014 8:50:36 PM
$sosrst.xvd	105,676	Regular File	1/20/2014 8:55:44 PM
$UpCase	128	Regular File	1/20/2014 8:50:36 PM
$Volume	0	Regular File	1/20/2014 8:50:36 PM
appswapfile.xvd	2,109,584	Regular File	1/20/2014 8:55:44 PM
AppTempStorage	3,164,364	Regular File	1/20/2014 8:55:44 PM
AppUserStorage	1,591,508	Regular File	1/20/2014 8:55:40 PM
ConnectedStorage-retail	9,548,892	Regular File	1/20/2014 8:55:40 PM
GDVRIndex.xvd	103,628	Regular File	1/20/2014 8:54:29 PM

Fig. 1. Temp content partition root directory after factory restoration.

Name	Size	Type	Date Modified
$Extend	1	Directory	1/20/2014 8:50:36 PM
$AttrDef	3	Regular File	1/20/2014 8:50:36 PM
$BadClus	0	Regular File	1/20/2014 8:50:36 PM
$Bitmap	1,312	Regular File	1/20/2014 8:50:36 PM
$Boot	8	Regular File	1/20/2014 8:50:36 PM
$I30	4	NTFS Index All...	1/22/2014 4:26:57 PM
$LogFile	65,536	Regular File	1/20/2014 8:50:36 PM
$MFT	256	Regular File	1/20/2014 8:50:36 PM
$MFTMirr	4	Regular File	1/20/2014 8:50:36 PM
$Secure	1	Regular File	1/20/2014 8:50:36 PM
$sosrst.xvd	143,076	Regular File	1/22/2014 4:58:00 PM
$UpCase	128	Regular File	1/20/2014 8:50:36 PM
$Volume	0	Regular File	1/20/2014 8:50:36 PM
appswapfile.xvd	2,109,584	Regular File	1/22/2014 4:55:31 PM
AppTempStorage	3,164,364	Regular File	1/22/2014 4:58:21 PM
AppUserStorage	1,591,508	Regular File	1/22/2014 4:56:13 PM
ConnectedStorage-retail	9,548,892	Regular File	1/22/2014 4:58:24 PM
GameDVR_25332748535625...	70,828	Regular File	1/22/2014 4:27:34 PM
GDVRIndex.xvd	103,628	Regular File	1/22/2014 4:28:45 PM
temp00	2,109,584	Regular File	1/22/2014 2:45:30 PM
temp01	2,109,584	Regular File	1/22/2014 4:13:08 PM

Fig. 2. Temp content partition root directory after scenarios.

respectively, on 01/21/2014. This may indicate that some feature of the system was used to recreate these files. Both files were created at a time that correlated with the initialization of playing Battlefield 4 in multiplayer mode over Xbox Live. It was also observed that the date modified of both files occurred within three minutes of when Dead Rising 3 started to be played in multiplayer mode.

On this basis, it seems likely that these files relate to connecting to Xbox Live and playing games with other users, so for instance, they may contain gamertags of other Xbox Live users who had played games/chatted with player using the Xbox One. Note that *$sosrt.xvd* changed during the scenario and was likely encrypted during Phase II, as the entropy scores changed between phases as shown in Table 2. It is possible that *$sosrt.xvd* is unencrypted until the first time the Xbox One device connects to Xbox Live, and that a private key associated with the user's Xbox Live account is then used to encrypt the file.

ConnectedStorage-retail was modified at shutdown in both phases. Its other timestamps did not help in discerning its function, but based on its name we hypothesize that this file is associated with connecting a memory device, such as a flash drive to the Xbox's USB port, or a planned expansion for the console.

Determining the functions of *AppTempStorage and AppUserStorage* proved to be difficult. *AppUserStorage* was last modified at shutdown in Phase I, and when FXNow was closed in Phase II. As a result of this, and the fact that the application files themselves (see Section System support partition) did not change when the respective application was used, this led us to hypothesize that *AppUserStorage* held user data for each application. For example, the Skype application requires the user to sign in with a login and password. This information can be saved so that it does not have to be entered each time the application is used; this file is where we hypothesize this type of data is held.

Table 2
Entropy scores for $sosrt.xvd.

File	Phase I Entropy Score	Phase II Entropy Score
$sosrt.xvd	4.160208895	7.89634165767

The *AppTempStorage* was updated in each phase at system shutdown. Based on its name, we hypothesized that the file contained temporary information for applications, which ensured that data would not be lost due to an unexpected shutdown.

Three new files were created in this partition during Phase II, *GameDVR_25332748 … 15f2.xvd*, *temp00*, and *temp01*.

The *GameDVR_25 … f2.xvd* file was not difficult to decipher. The file name itself pointed towards the Game DVR function of the Xbox One, and the date modified seemed to confirm this suspicion, as 4:28 PM on 1/22/2014 was the last time a game clip was recorded to the console. Its entropy score was 7.83 bits per byte, suggesting a compressed format such as compressed video, consistent with our expectations.

The names of the *temp00* and *temp01* files were not as clear as the *GameDVR_25..f2.xvd* file, but the timestamp of each file's creation found in the MFT suggests the function of both files. The Xbox One backs up data of video games on the cloud, so for instance if a user were to play a game on a friend's console, they would not lose any progress that was made when they continue playing on their own system. When a game is started, the system checks the cloud to determine if synchronization is necessary. Due to the factory reset, cloud synchronization was necessary and occurred the first time that both games were started during Phase II of the research. The modification times seen in Fig. 2 of *temp00* and *temp01* align with these events, for Battlefield 4 and Dead Rising 3 respectively.

User content partition

The contents of the root directory of the user content partition are shown in Figs. 3 and 4.

The files of interest were the last 8 files in Fig. 4, whose file names are strings of hexadecimal digits. As can be seen, the file names gave no clue to what the actual file was in this instance, however, there were a couple of aspects of the metadata that were of particular importance, namely the file sizes and the timestamps.

Due to the size of the installed games being freely available from Microsoft, the two chosen games (Battlefield 4 and Dead Rising 3) could be assumed to be the files of 35,981,096 bytes and 25,992,304 bytes respectively. This led to the hypothesis that the remaining six files were in fact the applications or settings that were downloaded during scenario testing. Information on the size of the

Name	Size	Type	Date Modified
$Extend	1	Directory	1/20/2014 8:50:36 PM
$AttrDef	3	Regular File	1/20/2014 8:50:36 PM
$BadClus	0	Regular File	1/20/2014 8:50:36 PM
$Bitmap	11,680	Regular File	1/20/2014 8:50:36 PM
$Boot	8	Regular File	1/20/2014 8:50:36 PM
$I30	4	NTFS Index All...	1/20/2014 8:50:36 PM
$LogFile	65,536	Regular File	1/20/2014 8:50:36 PM
$MFT	256	Regular File	1/20/2014 8:50:36 PM
$MFTMirr	4	Regular File	1/20/2014 8:50:36 PM
$Secure	1	Regular File	1/20/2014 8:50:36 PM
$UpCase	128	Regular File	1/20/2014 8:50:36 PM
$Volume	0	Regular File	1/20/2014 8:50:36 PM

Fig. 3. User content partition root directory after factory restoration.

Name	Size	Type	Date Modified
$Extend	1	Directory	1/20/2014 8:50:36 PM
$AttrDef	3	Regular File	1/20/2014 8:50:36 PM
$BadClus	0	Regular File	1/20/2014 8:50:36 PM
$Bitmap	11,680	Regular File	1/20/2014 8:50:36 PM
$Boot	8	Regular File	1/20/2014 8:50:36 PM
$I30	4	NTFS Index All...	1/20/2014 8:50:36 PM
$LogFile	65,536	Regular File	1/20/2014 8:50:36 PM
$MFT	256	Regular File	1/20/2014 8:50:36 PM
$MFTMirr	4	Regular File	1/20/2014 8:50:36 PM
$Secure	1	Regular File	1/20/2014 8:50:36 PM
$UpCase	128	Regular File	1/20/2014 8:50:36 PM
$Volume	0	Regular File	1/20/2014 8:50:36 PM
13096BD0-8237-47FA-80BE-...	59,652	Regular File	1/21/2014 3:18:54 PM
168859A8-2F07-4C63-9F3A-...	35,981,096	Regular File	1/21/2014 5:25:09 A...
242BF9CE-DA7C-4872-805E...	48,708	Regular File	1/21/2014 2:44:42 PM
508CC49E-41EC-4836-B927...	27,512	Regular File	1/22/2014 4:48:36 PM
B0655109-C128-4519-9E36-...	38,452	Regular File	1/21/2014 2:43:20 PM
B72D5AA7-6941-472A-8A5...	42,556	Regular File	1/22/2014 4:42:27 PM
D0134385-33C0-4382-BE31-...	22,036	Regular File	1/21/2014 2:46:40 PM
FD10657E-CB08-455B-A0D3...	25,992,304	Regular File	1/22/2014 4:09:29 PM

Fig. 4. User content partition root directory after scenarios.

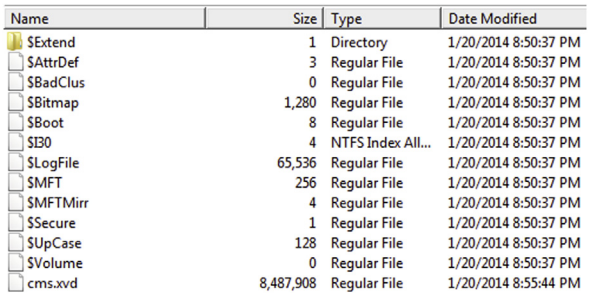
applications was not freely available as it was with the video games, therefore, further conclusions could not be reached without more information. After parsing the MFT for this partition, we obtained the timestamps for these smaller files. This allowed us to match files with corresponding game/application installations/executions in our scenario by timestamps if not by file size. Our mapping is shown in Table 3.

Conversely, the timestamps of file modification for Battlefield 4 and Dead Rising 3 did not coincide with the dates they were installed. This led us to hypothesize that the games write to the disk periodically when played (e.g. while saving the player's progress), thereby changing the file modification timestamp of their corresponding file. This is supported by the observation that their file modification times coincided with time that each game was last played. The creation timestamps were consistent with installation times of each game, consistent with our identification of each file by the known size of the game's installation footprint.

These files had entropy scores consistent with encrypted and/or compressed data (7.99 bits by byte). We believe the files are likely heavily compressed to conserve space,

Table 3
File name with its corresponding usage scenario.

File name	Related game/ Application
168859A8-2F07-4C63-9F3A-B89D056B6239	Battlefield 4
B0655109-C128-4519-9E36-0D370809CD0E	Xbox Video
242BF9CE-DA7C-4872-805E-E873ADB32C07	Skype
D0134385-33C0-4382-BE31-58C4CF4F453E	Twitch
13096BD0-8237-47FA-80BE-29A3563CF0BF	YouTube
FD10657E-CB08-455B-A0D3-0088CC93EAED	Dead Rising 3
B72D5AA7-6941-472A-8A5C-8BACE4D0B6DF	Xbox Music
508CC49E-41EC-4836-B927-C941BEAF4D6E	FXNow



Name	Size	Type	Date Modified
$Extend	1	Directory	1/20/2014 8:50:37 PM
$AttrDef	3	Regular File	1/20/2014 8:50:37 PM
$BadClus	0	Regular File	1/20/2014 8:50:37 PM
$Bitmap	1,280	Regular File	1/20/2014 8:50:37 PM
$Boot	8	Regular File	1/20/2014 8:50:37 PM
$I30	4	NTFS Index All...	1/20/2014 8:50:37 PM
$LogFile	65,536	Regular File	1/20/2014 8:50:37 PM
$MFT	256	Regular File	1/20/2014 8:50:37 PM
$MFTMirr	4	Regular File	1/20/2014 8:50:37 PM
$Secure	1	Regular File	1/20/2014 8:50:37 PM
$UpCase	128	Regular File	1/20/2014 8:50:37 PM
$Volume	0	Regular File	1/20/2014 8:50:37 PM
cms.xvd	8,487,908	Regular File	1/20/2014 8:55:44 PM

Fig. 5. System support partition root directory after factory restoration.

and possibly encrypted to minimize the risk of software piracy.

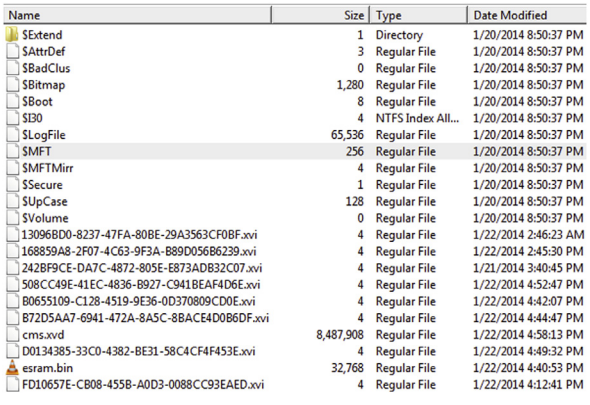
System support partition

There were several files to note in this partition, namely the *cms.xvd* file, the *esram.bin* file, and eight files with an xvi extension. (Figs. 5 and 6)

In Phase I the timestamp for the last time that *cms.xvd* was modified occurred when the system was shutdown. Similarly, in Phase II, the last modified timestamp aligned with the time that the system was shutdown. As shutdown was the only event that the timestamps for *cms.xvd* aligned, the actual contents of the file could not be hypothesized.

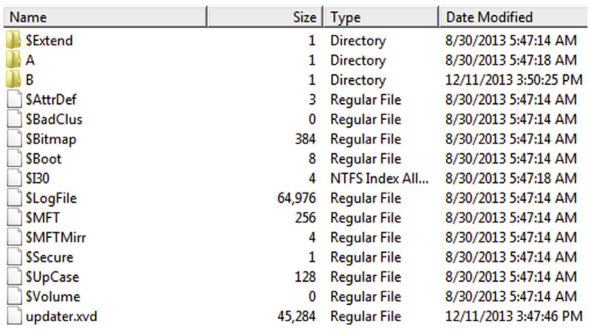
Due to its size, name, and the fact that the *esram.bin* file was not created until 1/22/2014 at 06:45:49, we hypothesize that it corresponded to the 32 MB of embedded static RAM (esRAM) storage found in the Xbox One. The entropy score of *esram.bin* was 6.7 bits per byte, a comparable entropy score to most JPEG graphics files reported by Shannon [16].

It was observed that eight of the files on this partition had the same file name as the downloaded applications and games found in the user content partition (see Fig. 4), but have a file extension of *.xvi*. Their file creation time corresponded to the time that each application/game was installed. We found no special headers for the *.xvi* file type, nor were the contents able to be read. Entropy scores were very low (e.g. *D0134385-33C0-4382-BE31-58C4CF4F453E.xvi* had an entropy score of just 0.08 bits per byte) – so low that we did not know what to make of them.



Name	Size	Type	Date Modified
$Extend	1	Directory	1/20/2014 8:50:37 PM
$AttrDef	3	Regular File	1/20/2014 8:50:37 PM
$BadClus	0	Regular File	1/20/2014 8:50:37 PM
$Bitmap	1,280	Regular File	1/20/2014 8:50:37 PM
$Boot	8	Regular File	1/20/2014 8:50:37 PM
$I30	4	NTFS Index All...	1/20/2014 8:50:37 PM
$LogFile	65,536	Regular File	1/20/2014 8:50:37 PM
$MFT	256	Regular File	1/20/2014 8:50:37 PM
$MFTMirr	4	Regular File	1/20/2014 8:50:37 PM
$Secure	1	Regular File	1/20/2014 8:50:37 PM
$UpCase	128	Regular File	1/20/2014 8:50:37 PM
$Volume	0	Regular File	1/20/2014 8:50:37 PM
13096BD0-8237-47FA-80BE-29A3563CF0BF.xvi	4	Regular File	1/22/2014 2:46:23 AM
168859A8-2F07-4C63-9F3A-B89D056B6239.xvi	4	Regular File	1/22/2014 2:45:30 PM
242BF9CE-DA7C-4872-805E-E873ADB32C07.xvi	4	Regular File	1/21/2014 3:40:45 PM
508CC49E-41EC-4836-B927-C941BEAF4D6E.xvi	4	Regular File	1/22/2014 4:52:47 PM
B0655109-C128-4519-9E36-0D370809CD0E.xvi	4	Regular File	1/22/2014 4:42:07 PM
B72D5AA7-6941-472A-8A5C-8BACE4D0B6DF.xvi	4	Regular File	1/22/2014 4:44:47 PM
cms.xvd	8,487,908	Regular File	1/22/2014 4:58:13 PM
D0134385-33C0-4382-BE31-58C4CF4F453E.xvi	4	Regular File	1/22/2014 4:49:32 PM
esram.bin	32,768	Regular File	1/22/2014 4:40:53 PM
FD10657E-CB08-455B-A0D3-0088CC93EAED.xvi	4	Regular File	1/22/2014 4:12:41 PM

Fig. 6. System support partition root directory after scenarios.



Name	Size	Type	Date Modified
$Extend	1	Directory	8/30/2013 5:47:14 AM
A	1	Directory	8/30/2013 5:47:18 AM
B	1	Directory	12/11/2013 3:50:25 PM
$AttrDef	3	Regular File	8/30/2013 5:47:14 AM
$BadClus	0	Regular File	8/30/2013 5:47:14 AM
$Bitmap	384	Regular File	8/30/2013 5:47:14 AM
$Boot	8	Regular File	8/30/2013 5:47:14 AM
$I30	4	NTFS Index All...	8/30/2013 5:47:18 AM
$LogFile	64,976	Regular File	8/30/2013 5:47:14 AM
$MFT	256	Regular File	8/30/2013 5:47:14 AM
$MFTMirr	4	Regular File	8/30/2013 5:47:14 AM
$Secure	1	Regular File	8/30/2013 5:47:14 AM
$UpCase	128	Regular File	8/30/2013 5:47:14 AM
$Volume	0	Regular File	8/30/2013 5:47:14 AM
updater.xvd	45,284	Regular File	12/11/2013 3:47:46 PM

Fig. 7. System update partition both before and after use.

Therefore, the function of these particular files was not determined. These files will be included in the data set that will be released (see Section Contribution).

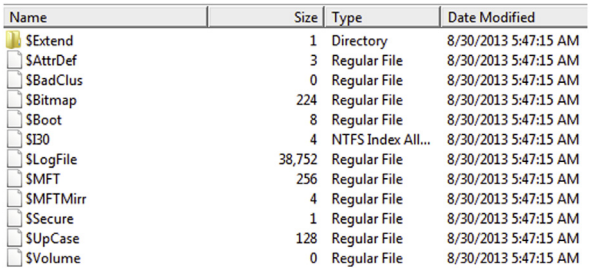
System update and system update 2 partition

Both of these partitions showed alternations only to existing files between Phase I and Phase II, with no new files created. Figs. 7 and 8 display the partition's root contents.

This Xbox One was purchased on 11/30/2013 and was first set up by the user on 12/3/2013. *Directories A* and *B* were both created on the same time and day, 8/30/2013 05:47:18. *Directory A* contained six files that had a modification timestamp on 12/3/2013 with times ranging from 10:12:20 to 10:15:33 PM, a date and time that coincided with the initial set up of the system by the user. *Directory B*, its contents, and *updater.xvd* were last modified on 12/11/2013 between times 3:47:46 and 3:51:14 PM, a date and times that corresponded to the latest system update received by the Xbox One. We concluded that these files therefore related to system configuration and updates and were unlikely to contain user data.

XVD files

Some of the partitions, mainly in *directories A* and *B* of the system update partition (Figs. 9 and 10), contained files with an extension of *xvd*. Other appearances of the *xvd* extension were seen with the files: *cms.xvd*, *$sosrst.xvd*, *appswapfile.xvd*, *GameDVR.xvd*, *GDVRIndex.xvd*, and *updater.xvd*.



Name	Size	Type	Date Modified
$Extend	1	Directory	8/30/2013 5:47:15 AM
$AttrDef	3	Regular File	8/30/2013 5:47:15 AM
$BadClus	0	Regular File	8/30/2013 5:47:15 AM
$Bitmap	224	Regular File	8/30/2013 5:47:15 AM
$Boot	8	Regular File	8/30/2013 5:47:15 AM
$I30	4	NTFS Index All...	8/30/2013 5:47:15 AM
$LogFile	38,752	Regular File	8/30/2013 5:47:15 AM
$MFT	256	Regular File	8/30/2013 5:47:15 AM
$MFTMirr	4	Regular File	8/30/2013 5:47:15 AM
$Secure	1	Regular File	8/30/2013 5:47:15 AM
$UpCase	128	Regular File	8/30/2013 5:47:15 AM
$Volume	0	Regular File	8/30/2013 5:47:15 AM

Fig. 8. System update 2 partition both before and after use.

Name	Size	Type	Date Modified
$I30	4	NTFS Index All...	8/30/2013 5:47:18 AM
deltas.xvd	349,356	Regular File	12/3/2013 10:14:00 PM
ExtraSettings.xvd	16,500	Regular File	5/25/2013 4:02:38 PM
SettingsTemplate.xvd	37,144	Regular File	12/3/2013 10:12:39 PM
sosinit.xvd	24,204	Regular File	12/3/2013 10:12:20 PM
sostmpl.xvd	63,516	Regular File	12/3/2013 10:12:28 PM
system.xvd	870,596	Regular File	12/3/2013 10:15:13 PM
systemaux.xvd	273,876	Regular File	12/3/2013 10:15:33 PM

Fig. 9. Directory A of system update partition.

The string of *msft-xvd* was observed in every *xvd* file at addr 0x200. The other content was non-human readable, although the entropy scores of various xvd files suggest that not every *xvd* file is actually the same file type. The format may simply be a "wrapper" around other binary data. Occasionally, we noted that the entropy scores changed between Phase I and Phase II, showing that an encryption or compression takes place after the user connects to Xbox Live or during use in Phase II. The entropy scores for a collection of *xvd* files are shown for each phase in Table 4.

There is much speculation in the Xbox modding community as to what the *xvd* file type is. Modders have been examining this file type since the release of the system and have advanced a variety of opinions as to its function. Some believe that these files are modified Windows Imaging Format (WIM) files, others believe that it is a brand-new, custom format developed by Microsoft, though, the majority of the community believes the files are package files, a much more complex and secured version of the Xbox 360 Secure Transacted File System (STFS) packages (HorizonMB, 2013). The STFS was the file system used by the Xbox 360 for all packages created and downloaded by the system (Free60, 2014). It is our opinion, based on entropy score disparity, that the Xbox One's *xvd* files actually belong to a variety of different types all with the same file extension.

We released an obtainable data set of these files available to the forensic community so that additional research may be conducted without the need for an Xbox One (see Section Contribution).

Network forensics

No passwords or user names were found when capturing the network traffic, however, we were able to discern when a user signed in due to the sequence of captured files as shown in Fig. 11. The file name highlighted in Fig. 11 did not appear in any other scenario tested, only when a user was signed in, which indicated that this certificate was used to verify user sign in.

It seemed that each application tested employed its own measure of security. For instance, Skype seemed to be fully encrypted. You could not see any of the messages sent or

Name	Size	Type	Date Modified
deltas.xvd	165,756	Regular File	12/11/2013 3:49:03 PM
SettingsTemplate.xvd	37,144	Regular File	12/11/2013 3:48:05 PM
sosinit.xvd	24,204	Regular File	12/11/2013 3:48:15 PM
sostmpl.xvd	63,516	Regular File	12/11/2013 3:48:36 PM
system.xvd	870,596	Regular File	12/11/2013 3:50:25 PM
systemaux.xvd	273,876	Regular File	12/11/2013 3:51:14 PM

Fig. 10. Directory B of system update partition.

Table 4
Rounded entropy scores (in bits per byte) of xvd files.

File name	Phase I Entropy Score	Phase II Entropy Score
$sosrst.xvd[a]	4.1602	7.8963
GDVRIndex.xvd	1.5438	1.5438
systemaux.xvd[a]	5.3353	7.8229
SettingsTemplate.xvd	7.5681	7.5681
deltas.xvd	7.6256	7.6256
sostmpl.xvd	6.5402	6.5402
system.xvd[a]	5.3031	7.9995
updater.xvd	7.8123	7.8123

[a] Entropy score changed between Phases I and II.

received, or hear any of the VOIP conversations. However, we were able to see exactly when Skype was started up by the Transport Layer Security (TLS) Certificate found, its timestamp correlated with the time that we had started Skype, but that was the extent of what could be seen.

On the other hand, the Twitch TV, YouTube, Game DVR, and Internet Explorer applications actually allowed us to view what the user was doing. It was possible to see when the user was on the Twitch TV application as well as the stream that they were viewing. This information could be discerned from the files tab of NetworkMiner as shown in Fig. 12. With this information, it was clear that the user was watching the *nightblue3* stream.

The YouTube application only allowed us to view a portion of the actual video that was viewed and the link found did not tie back to the exact URL of the viewed video. Therefore, determining the exact video watched may prove difficult to determine.

We were able to view the entire game clip that was watched on the Game DVR application. An image of this can be seen in Fig. 13.

The Internet Explorer application allowed us to capture data as if the user was browsing the web on their computer. Therefore, we were able to see exactly what the user did on any site that did not have ample security to prevent us from viewing the traffic.

When we investigated the network traffic of both games, Battlefield 4 and Dead Rising 3, we discovered that the network traffic was encrypted. We could not see the exact actions that took place, meaning we could not see the mode the user was playing in or who they were playing with, however we were able to tell what game was being

D. port	Protocol	Filename	Extension	Size
TCP 50...	TlsCertificate	licensing.xboxlive.com[10].cer	cer	1 569 B
TCP 50...	TlsCertificate	MSIT Machine Auth CA 2[10].cer	cer	1 548 B
TCP 50...	TlsCertificate	Microsoft Internet Author[10].cer	cer	1 285 B
TCP 50...	TlsCertificate	xboxlive.com.cer	cer	1 590 B
TCP 50...	TlsCertificate	MSIT Machine Auth CA 2.cer	cer	1 548 B
TCP 50...	TlsCertificate	Microsoft Internet Author.cer	cer	1 285 B
TCP 50...	TlsCertificate	accounts.xboxlive.com.cer	cer	1 568 B
TCP 50...	TlsCertificate	MSIT Machine Auth CA 2.cer	cer	1 548 B
TCP 50...	TlsCertificate	Microsoft Internet Author.cer	cer	1 285 B
TCP 50...	TlsCertificate	accounts.xboxlive.com[1].cer	cer	1 568 B
TCP 50...	TlsCertificate	MSIT Machine Auth CA 2[1].cer	cer	1 548 B
TCP 50...	TlsCertificate	Microsoft Internet Author[1].cer	cer	1 285 B
TCP 50...	TlsCertificate	userpresence.xboxlive.com[2].cer	cer	1 637 B
TCP 50...	TlsCertificate	MSIT Machine Auth CA 2[2].cer	cer	1 548 B
TCP 50...	TlsCertificate	Microsoft Internet Author[2].cer	cer	1 285 B
TCP 50...	TlsCertificate	userpresence.xboxlive.com[3].cer	cer	1 637 B

Fig. 11. Captured data when signing in.

	Details
28 AM	usher.justin.tv/api/channel/hls/nightblue3.m3u8?token={"user_id":null,"channel":"nightblue3","expires"
29 AM	video3.iad02.hls.twitch.tv/hls94/nightblue3_9310351552_87830761/low/index-live.m3u8?token=id=893
29 AM	video3.iad02.hls.twitch.tv/hls94/nightblue3_9310351552_87830761/high/index-live.m3u8?token=id=89
29 AM	video3.iad02.hls.twitch.tv/hls94/nightblue3_9310351552_87830761/medium/index-live.m3u8?token=id=
29 AM	video3.iad02.hls.twitch.tv/hls94/nightblue3_9310351552_87830761/low/index-0000006700-cRCH.ts
29 AM	video3.iad02.hls.twitch.tv/hls94/nightblue3_9310351552_87830761/mobile/index-live.m3u8?token=id=
29 AM	video3.iad02.hls.twitch.tv/hls94/nightblue3_9310351552_87830761/high/index-live.m3u8?token=id=89
29 AM	video3.iad02.hls.twitch.tv/hls94/nightblue3_9310351552_87830761/high/index-0000006701-1gPJ.ts
30 AM	video3.iad02.hls.twitch.tv/hls94/nightblue3_9310351552_87830761/high/index-0000006702-vgEF.ts
31 AM	video3.iad02.hls.twitch.tv/hls94/nightblue3_9310351552_87830761/high/index-0000006703-WWL9.ts
31 AM	video3.iad02.hls.twitch.tv/hls94/nightblue3_9310351552_87830761/high/index-0000006704-7h29.ts
32 AM	video3.iad02.hls.twitch.tv/hls94/nightblue3_9310351552_87830761/high/index-0000006705-4217.ts
33 AM	video3.iad02.hls.twitch.tv/hls94/nightblue3_9310351552_87830761/high/index-0000006706-Jp6l.ts
34 AM	video3.iad02.hls.twitch.tv/hls94/nightblue3_9310351552_87830761/high/index-live.m3u8?token=id=89
34 AM	video3.iad02.hls.twitch.tv/hls94/nightblue3_9310351552_87830761/high/index-0000006707-owXd.ts
38 AM	video3.iad02.hls.twitch.tv/hls94/nightblue3_9310351552_87830761/high/index-live.m3u8?token=id=89

Fig. 12. Captured data from Twitch TV.

played by the captured traffic. Battlefield 4 was distinguished by a TLS Certificate shown in Fig. 14. The initiation of Dead Rising caused a lot of network traffic, and although the content of the files was illegible they could be directly linked to Dead Rising by their file name. (Fig. 15)

Future work

More research is needed with regard to the Xbox One, its file system, and the encryption methods used. Understanding the file types (and potentially, sub-types) found within the Xbox One, such as *.xvd* and *.xvi*, is necessary to further examine the console. Although many of these files seem, on the basis of their entropy scores, to be encrypted (or at least compressed with an unknown scheme), some files were not, and these should therefore be priorities for reverse engineering.

It may also prove valuable to look at an Xbox One hard disk drive when it is first shipped. The hard drive employed in this experimental work had been used prior to selecting it. Although it was restored to factory settings, it may prove useful to see what, if anything, is dissimilar between a factory restored drive versus a newly shipped drive.

A more thorough examination of the Xbox Live network is needed. There are many other applications and games that need to be tested. Additionally, there is a mobile phone

Fig. 13. Captured video from Game DVR.

50...	TlsCertificate	Microsoft Internet Author[2].cer	cer	1 285 B	4/23/2014 12:47:11 PM
50...	TlsCertificate	bf4.gos.ea.com.cer	cer	1 364 B	4/23/2014 12:47:11 PM
50...	TlsCertificate	VeriSign Class 3 Secure S.cer	cer	1 520 B	4/23/2014 12:47:11 PM

Fig. 14. Captured data from Battlefield 4.

application known as Smart Glass that allows the user to interact with the Xbox through their phone. It would be of interest to examine this relationship to see what data can be found in the transmissions between the phone and Xbox One.

A database could be made containing all of the downloaded material available for the Xbox One, along with hash values and the file sizes of that material. This would allow for quick identification of different applications, games, or other downloaded material, thereby making investigations involving this device more efficient.

Conclusions

This research provides the initial foundation for understanding how an Xbox One can be examined in a forensically sound manner. Even without modifications, the Xbox One is a very powerful computing device and should not be seen as just a gaming console; it is marketed for its multimedia capabilities just as much as it is for its video games. As with the previous generations of video game consoles, crime will undoubtedly take place involving the Xbox One, both over Xbox Live and locally, thus making it essential that investigators understand how to analyze the system.

The complexity of the Xbox One seems greater than its predecessors. It appears to make heavy usage of encryption (at least after connection to Xbox Live), and its new file types made it extremely difficult to discern any information. We were not able to determine exactly when or how the user used applications, watching television did not seem to log any data on the system, and signing in to the user's profile could not be determined from the metadata alone. The data contained in the MFT along with information ascertained from this research can be of great importance to investigators. We were able to retrospectively link files to the games and applications that were installed on the Xbox One, we could see when the console was last shutoff, we could see when the system was restored to factory settings, and were even able to determine the first time the user ever used the system. The metadata found in the MFT can be used by investigators to develop timelines, patterns of use, and corroborate a story.

ırt	Protocol	Filename	Extension	Size	Timestamp	D
50...	HttpGetNormal	deadrising3_1.0.0.5_.txt	txt	4 096 B	4/23/2014 12:55:43 PM	ic
50...	HttpGetNormal	deadrising3_1.0.0.5_[1].txt	txt	4 096 B	4/23/2014 12:55:43 PM	ic
50...	HttpGetNormal	deadrising3_1.0.0.5_[2].txt	txt	4 096 B	4/23/2014 12:55:43 PM	ic
50...	HttpGetNormal	deadrising3_1.0.0.5_[3].txt	txt	32 768 B	4/23/2014 12:55:43 PM	ic
50...	HttpGetNormal	deadrising3_1.0.0.5_[4].txt	txt	4 096 B	4/23/2014 12:55:43 PM	ic
50...	HttpGetNormal	deadrising3_1.0.0.5_[5].txt	txt	32 768 B	4/23/2014 12:55:43 PM	ic
50...	HttpGetNormal	deadrising3_1.0.0.5_[6].txt	txt	4 096 B	4/23/2014 12:55:43 PM	ic
50...	HttpGetNormal	deadrising3_1.0.0.5_[7].txt	txt	16 384 B	4/23/2014 12:55:43 PM	ic
50...	HttpGetNormal	deadrising3_1.0.0.5_[8].txt	txt	4 096 B	4/23/2014 12:55:43 PM	ic
50...	HttpGetNormal	deadrising3_1.0.0.5_[9].txt	txt	4 096 B	4/23/2014 12:55:43 PM	ic
50...	HttpGetNormal	deadrising3_1.0.0.5_[10].txt	txt	4 096 B	4/23/2014 12:55:44 PM	ic
50...	HttpGetNormal	deadrising3_1.0.0.5_[11].txt	txt	4 096 B	4/23/2014 12:55:44 PM	ic
50...	HttpGetNormal	deadrising3_1.0.0.5_[12].txt	txt	4 096 B	4/23/2014 12:55:44 PM	ic
50...	HttpGetNormal	deadrising3_1.0.0.5_[13].txt	txt	32 768 B	4/23/2014 12:55:44 PM	ic
50...	HttpGetNormal	deadrising3_1.0.0.5_[14].txt	txt	32 768 B	4/23/2014 12:55:44 PM	ic
50...	HttpGetNormal	deadrising3_1.0.0.5_[15].txt	txt	4 096 B	4/23/2014 12:55:44 PM	ic
50...	HttpGetNormal	deadrising3_1.0.0.5_[16].txt	txt	4 096 B	4/23/2014 12:55:44 PM	ic
50...	HttpGetNormal	deadrising3_1.0.0.5_[17].txt	txt	4 096 B	4/23/2014 12:55:44 PM	ic

Fig. 15. Captured data from Dead Rising 3.

We also noted that the network traffic that was captured varied based on what specific applications were being used. Each application had different levels of security associated with them, for example we were not able to see the calls or messages made with Skype but we were able to view the exact game clip that was watched during the use of the Game DVR application. The video games themselves were secure. We were able to discern which game was being played but the actions that took place during play were not discovered.

References

AccessData. Product downloads. Retrieved from AccessData, http://www.accessdata.com/support/product-downloads; 2013.

Bolt S. XBOX 360 forensics: a digital forensics guide to examining artifacts. Syngress Publishing; 2011.

Buchholz F, Spafford E. On the role of file system metadata in digital forensics. Digit Investig 2004;1:298–309.

Carrier B. Projects. Retrieved from The Sleuth Kit, http://www.sleuthkit.org/; 2013.

Collins D. XFT: a forensic toolkit for the original Xbox game console. Int J Electron Secur Digital Forensics 2009;2(2):199–205.

Combs G. Wireshark. Retrieved from Wireshark, http://www.wireshark.org/; 2013.

Conrad S, Dorn G, Craiger JP. Forensic analysis of a Sony Play Station 3 gaming console. Retrieved from Computer Forensics LLC, http://consoleforensics.com/wpcontent/uploads/2009/12/2009_Dorn_et_al_PS3.pdf; 2009, December.

Free60. STFS. Retrieved February 2, 2014, from Free60, http://www.free60.org/STFS; 2014, January 7.

Garfinkel SL. Carving contiguous and fragmented files with fast object validation. Digit Investig 2007;4:2–12.

Garfinkel SL. Digital media triage with bulk data analysis and bulk_extractor. Comput Secur; 2012:56–72.

HorizonMB. [Community Research] Xbox One & Modding. Retrieved January 20, 2014, from HorisonMB, https://www.horizonmb.com/threads/156291-Community-Research-Xbox-One-amp-Modding; 2013, November 30.

Kent K, Chevalier S, Grance T, Dang H. Guide to integrating forensic techniques into incident response. Gaithersburg: NIST; 2006.

Modfreakz. HDDHACKR v1.40 Build 20130303. Retrieved from Xbox-Hacker BBS, http://www.xboxhacker.org/index.php?PHPSESSID=3f54baba38a3708fee576f6f7fa2a8ab&topic=11813.msg77053#msg77053; 2009, May 23.

NETRESEC. NetworkMiner. Retrieved from NETRESEC, http://www.netresec.com/?page=NetworkMiner; 2013.

Podhradsky D, Sley L, D'Ovidio D, Casey C. Identity theft and used gaming consoles: Recovering personal information from Xbox 360 hard drives. AMCIS 2011 Proceedings; 2011.

Richard III GG, Roussev V. Scalpel: A Frugal, high performance file Carver. Digital Forensic Research Workshop. New Orleans, LA; 2005.

Schicht J. mft2csv. Retrieved February 1, 2014, from mft2csv, https://code.google.com/p/mft2csv/downloads/detail?name=mft2csv_v2.0.0.13.zip&can=2&q=; 2014, January 23.

Shannon M. Forensic relative strength scoring: ASCII and entropy scoring. Int J Digit Evid 2004;2(4):151–69.

Vaughan C. Xbox security issues and forensic recovery methodology (utilising Linux). Digit Investig 2004;1:165–72.

Xynos K, Harries S, Sutherland I, Davies G, Blyth A. Xbox 360: a digital forensic investigation of the hard disk drive. Digit Investig 2010;6:104–11.